Cross Link Insertion for Improving Tolerance to Variations in Clock Network Synthesis

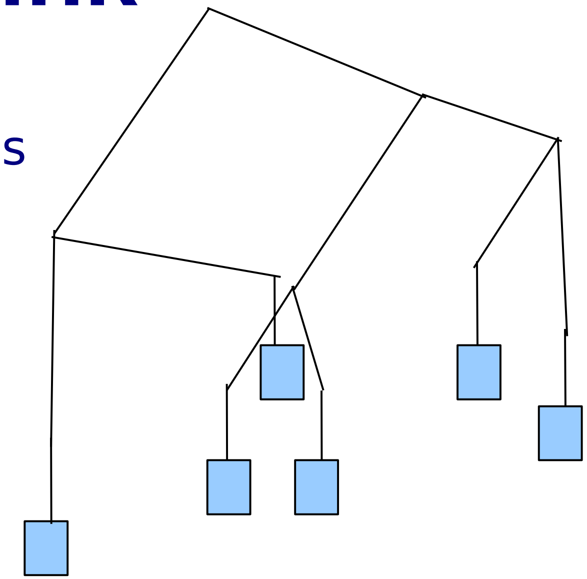
Tarun Mittal Cheng-Kok Koh
School of Electrical and Computer Engineering
Purdue University

Presentation Flow

- Introduction
- Comparison of link insertion schemes
- Clock Network Synthesis
- Experimental Results
- Conclusions and Future Work

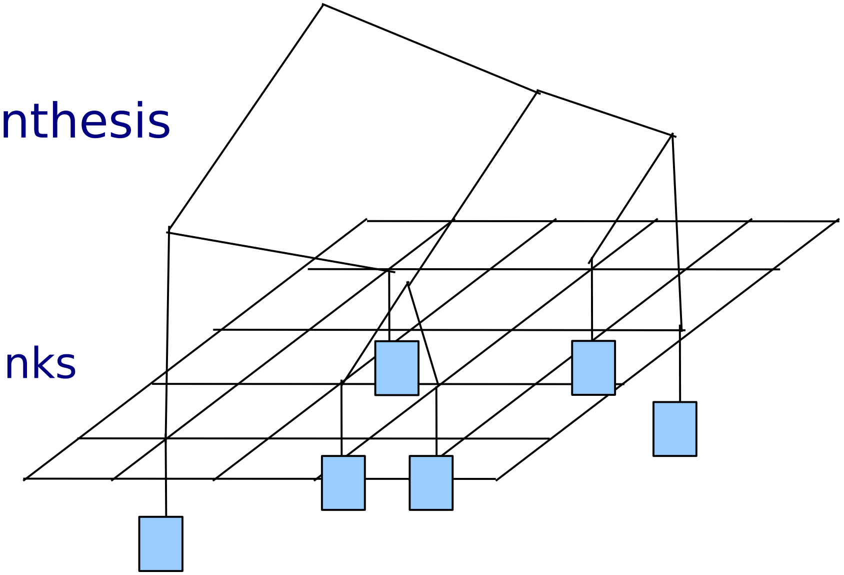
Insertion of Cross link

- Current approach to Clock Network Synthesis
  - Clock Trees
    - Shorter Wiring
    - Unique path from source to sinks
    - More susceptible to process variations



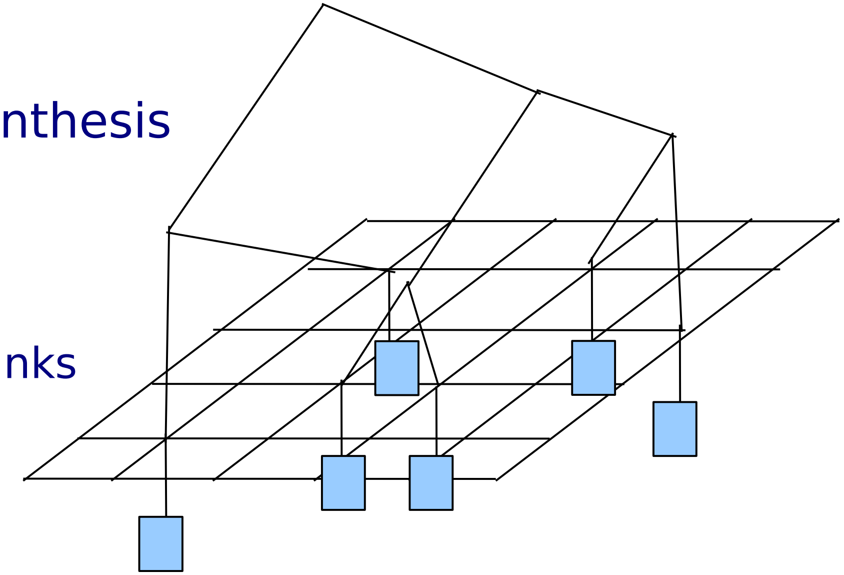
Insertion of Cross link

- Current approach to Clock Network Synthesis
  - Clock Trees
    - Shorter Wiring
    - Unique path from source to sinks
    - More susceptible to process variations
  - Clock Mesh
    - Higher wiring cost
    - Many paths from source to sinks
    - More robust to process variations

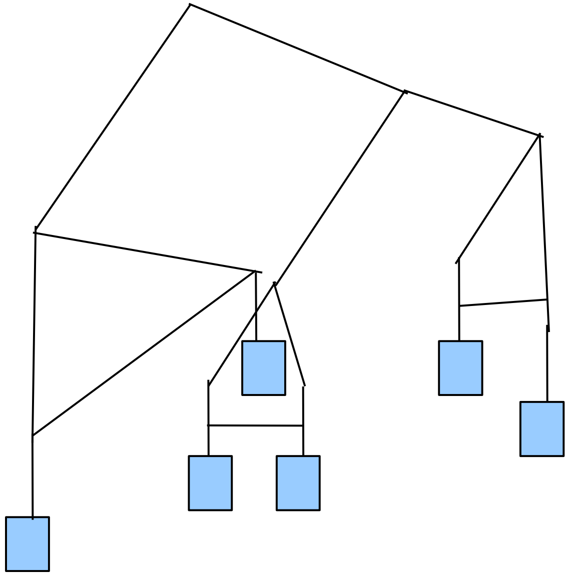


Insertion of Cross link

- Current approach to Clock Network Synthesis
  - Clock Trees
    - Shorter Wiring
    - Unique path from source to sinks
    - More susceptible to process variations
  - Clock Mesh
    - Higher wiring cost
    - Many paths from source to sinks
    - More robust to process variations



- Cross link form a compromise between clock trees and clock meshes



Effect of cross link insertion

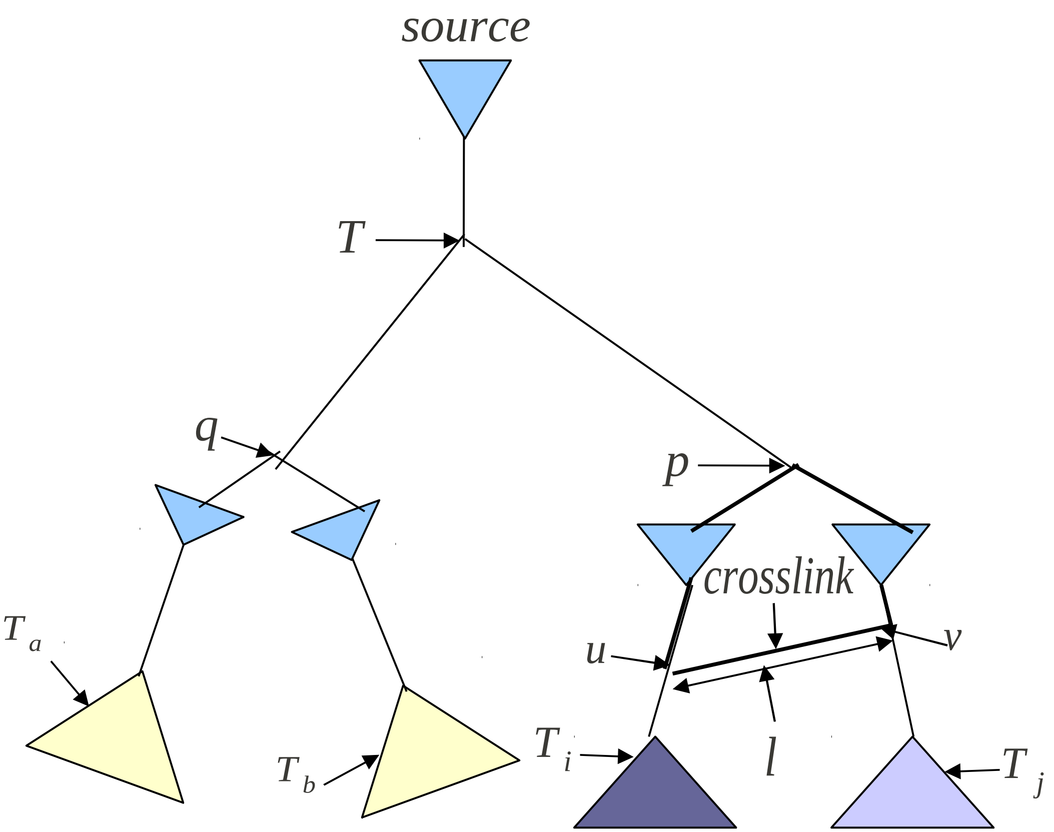
Change in skew between nodes u and v due to cross link addition

$$\overline{q}_{u,v} = \alpha q_{u,v} + \alpha\beta$$

where

$\overline{q}_{u,v}$=skew after link addition

$q_{u,v}$=skew before link addition



Effect of cross link insertion

Change in skew between nodes u and v due to cross link addition

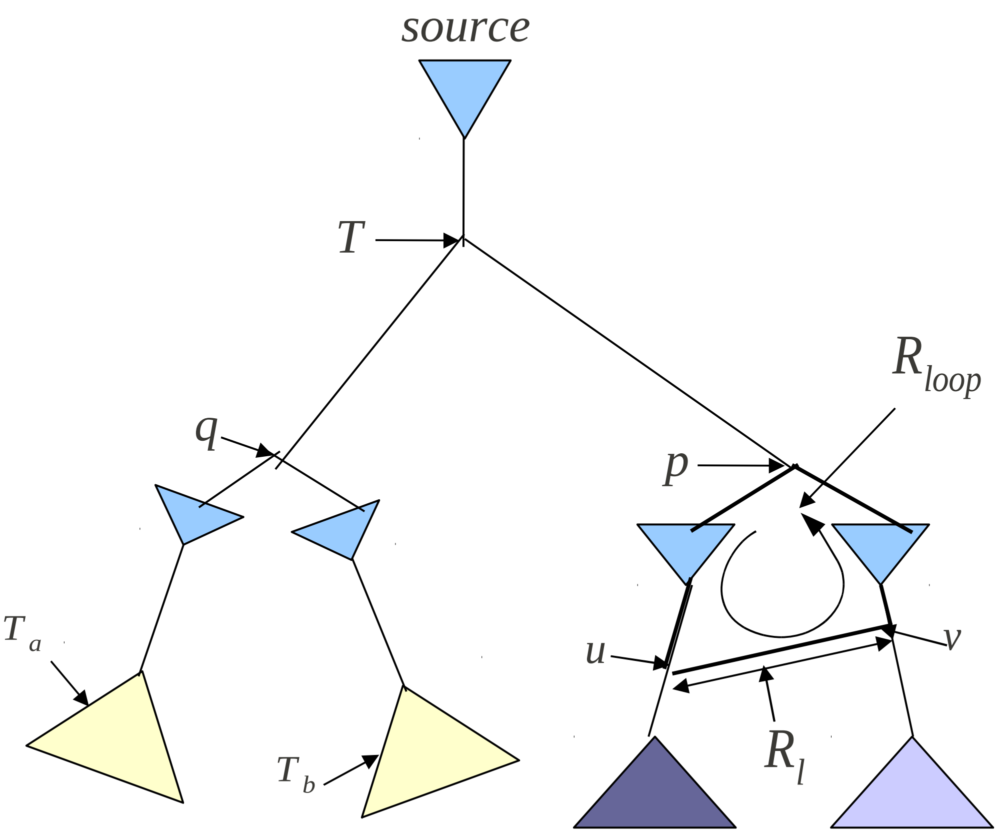
$$\overline{q}_{u,v} = \alpha q_{u,v} + \alpha\beta$$

where

$\overline{q}_{u,v}$=skew after link addition

$q_{u,v}$=skew before link addition

$\alpha = R_l / R_{loop}$



Effect of cross link insertion

Change in skew between nodes u and v due to cross link addition

$$\overline{q}_{u,v}= \alpha q_{u,v} + \alpha\beta$$

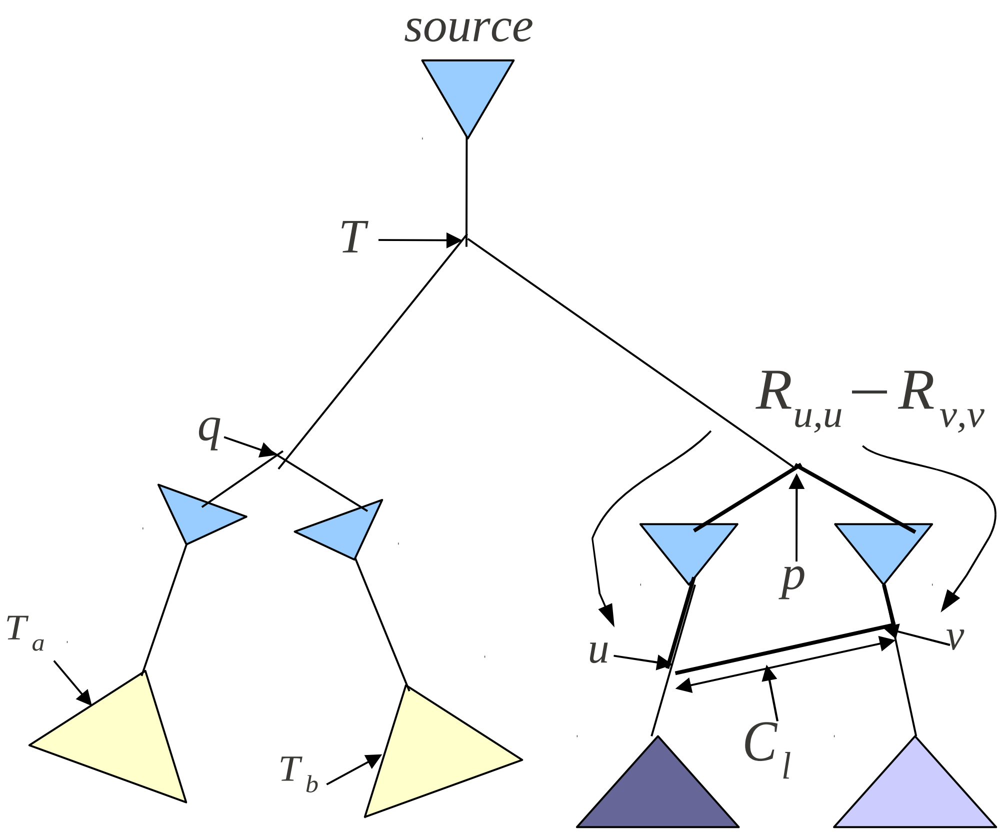
where

$\overline{q}_{u,v}$=skew after link addition

$q_{u,v}$=skew before link addition

$\alpha=R_l/R_{loop}$

$\beta=Cl/2(R_{u,u}-R_{v,v})$

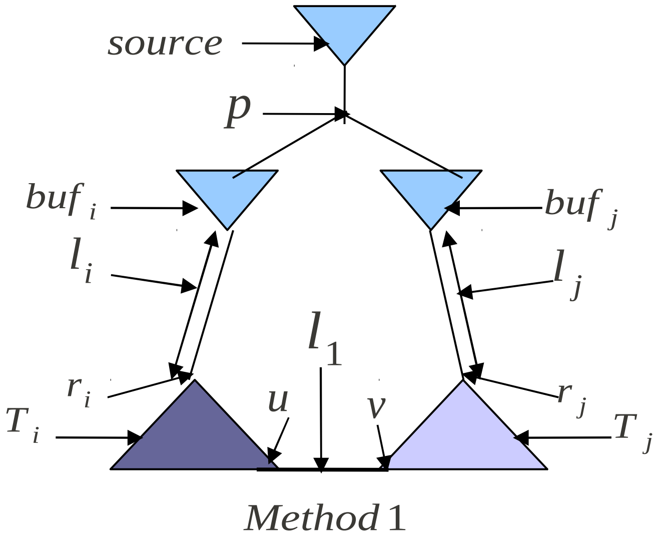


Comparison of Link insertion schemes

- Method 1:

  - Link l_1 is inserted between two sinks u and v

  - This method of link insertion is used in [Rajaram-Hu, ISPD'05]



Comparison of Link insertion schemes

- Method 1:

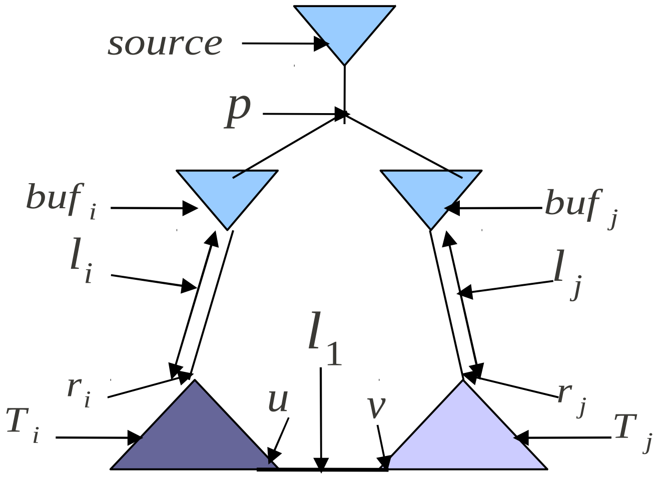
  - Link l_1 is inserted between two sinks u and v

  - This method of link insertion is used in [Rajaram-Hu, ISPD'05]

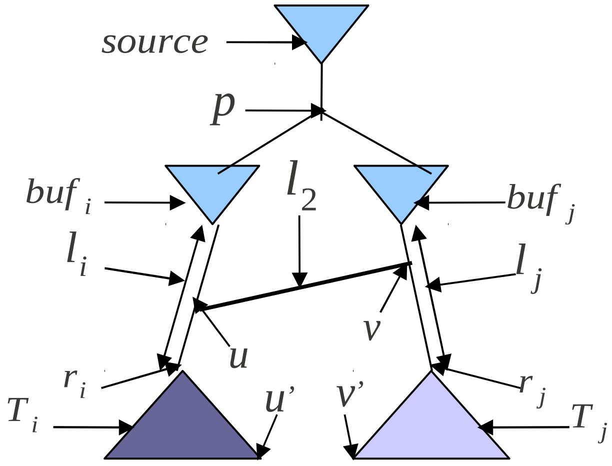
- Method 2:

  - Link l_2 is inserted between two higher level internal nodes u and v

  - This method of link insertion is used in our approach



Method 1



Method 2

Comparison of Link insertion schemes

- Method 1:

  - Link l_1 is inserted between two sinks u and v

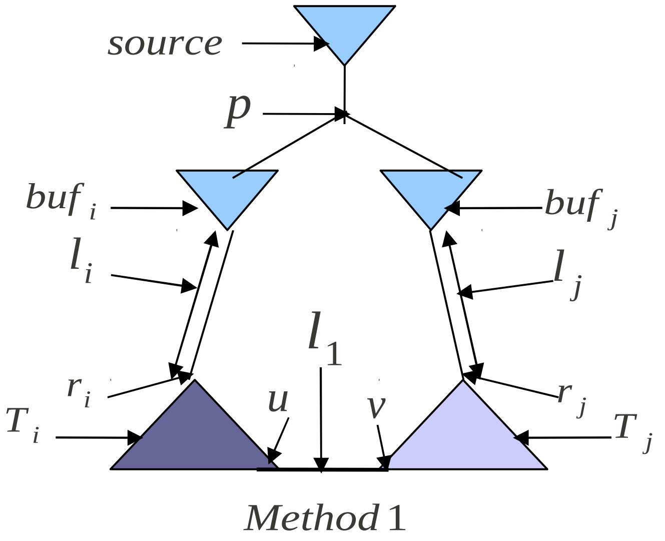
  - This method of link insertion is used in [Rajaram-Hu, ISPD'05]

- Method 2:

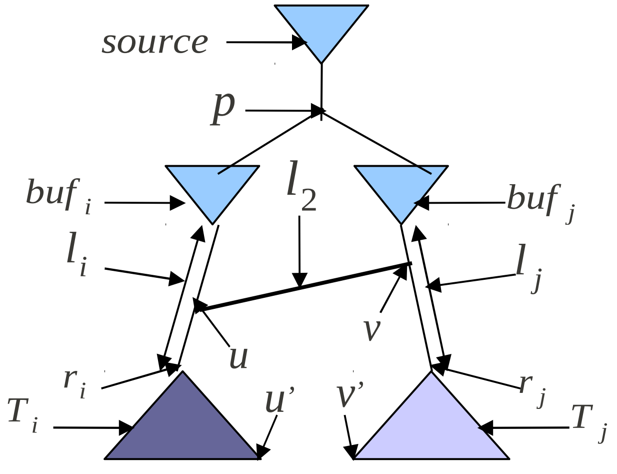
  - Link l_2 is inserted between two higher level internal nodes u and v

  - This method of link insertion is used in our approach

- $l_2 << l_1$ satisfies $\alpha_2<\alpha_1$ & $\beta_2<\beta_1$



Method 1

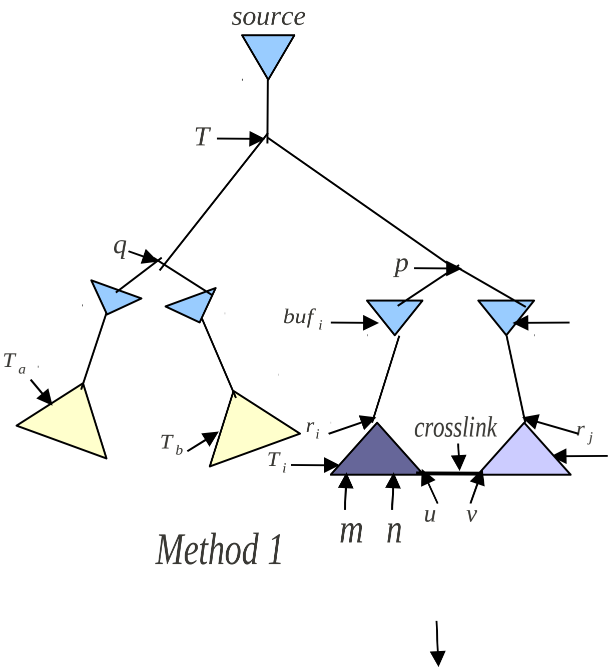


Method 2

Effect of cross link on sink delays

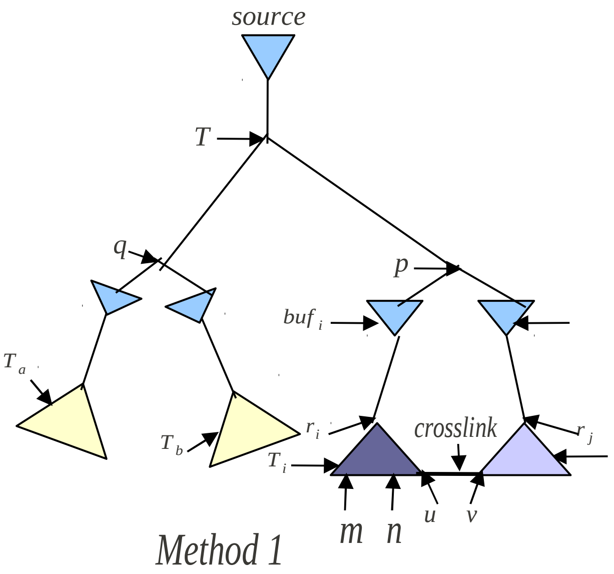
Sinks are in the same subtree

- Method 1:
  - m and n have different path lengths to the end point of the cross link
  - skew variability depends upon locality of sink node to the end point of the cross link

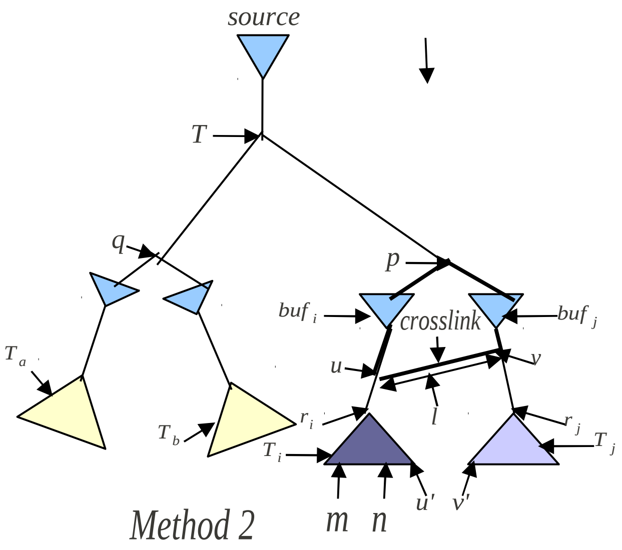


Sinks are in the same subtree

- Method 1:
  - m and n have different path lengths to the end point of the cross link
  - skew variability depends upon locality of sink node to the end point of the cross link



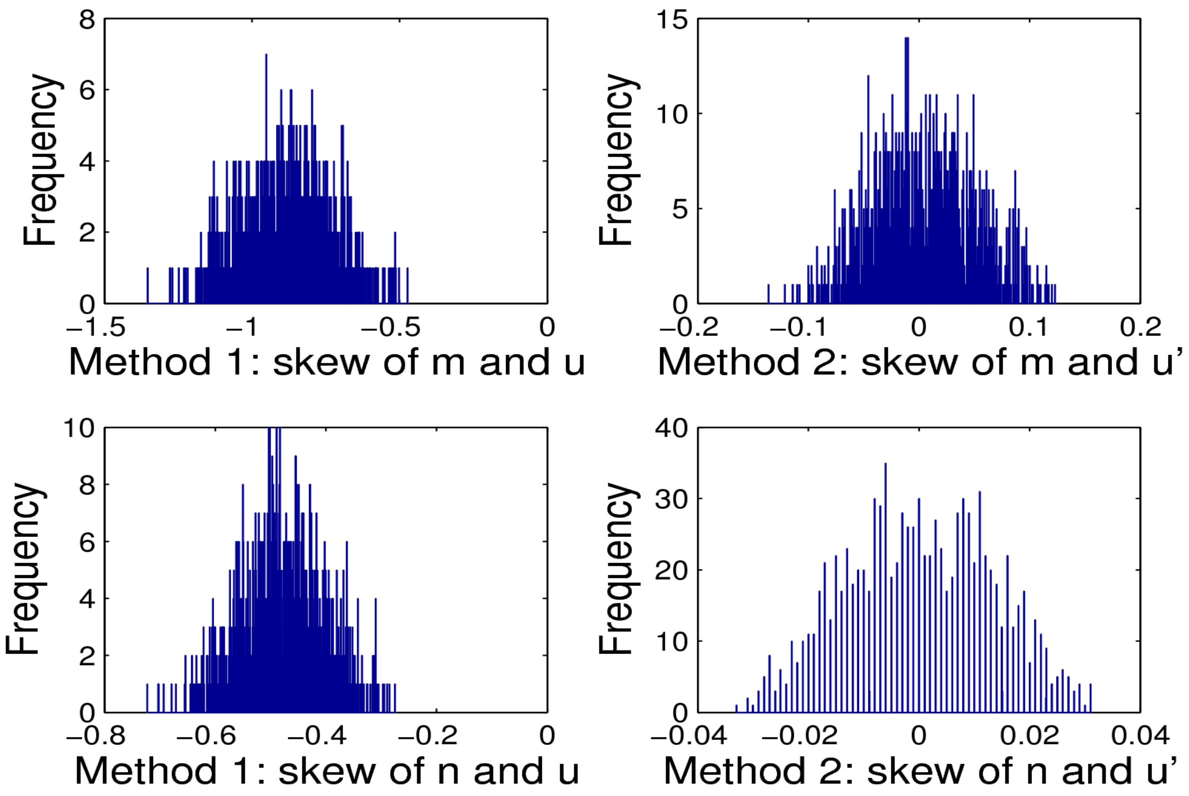
- Method 2:
  - m and n have nearly same path lengths to the end point of cross link
  - skew variability is same for the sink nodes



Measured skew variability for both methods

Range is 0 .75ps

Range is 0 .2ps

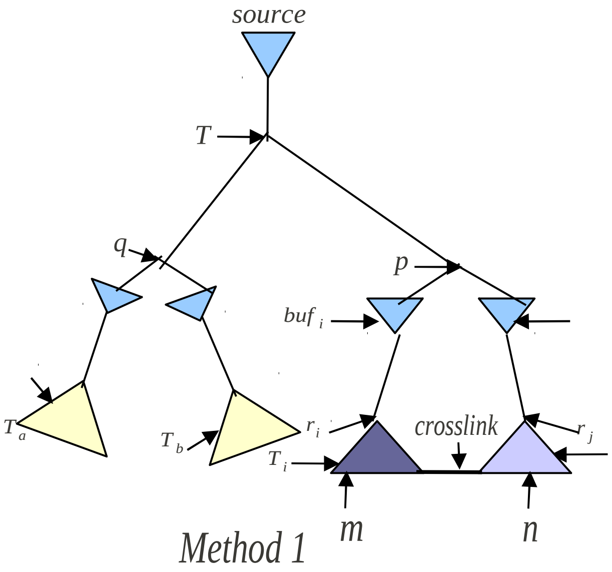


Range is 0 .4ps

Range is 0 .06ps

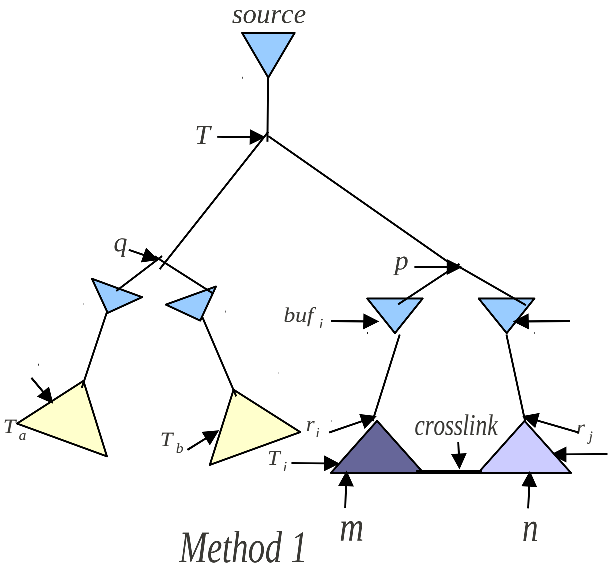
Sinks are in different sub-trees connected by the cross link

- Method 1:
  - Different delays for sinks within a sub-tree
  - Non uniform correlation between the sink pairs m and n

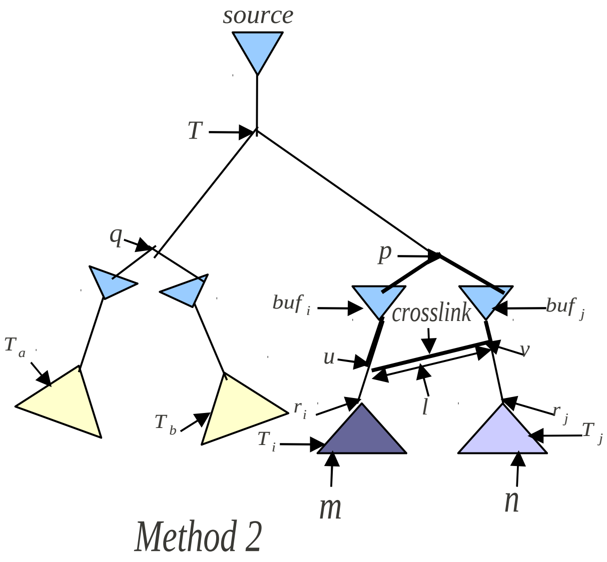


Sinks are in different sub-trees connected by the cross link

- Method 1:
  - Different delays for sinks within a sub-tree
  - Non uniform correlation between the sink pairs m and n

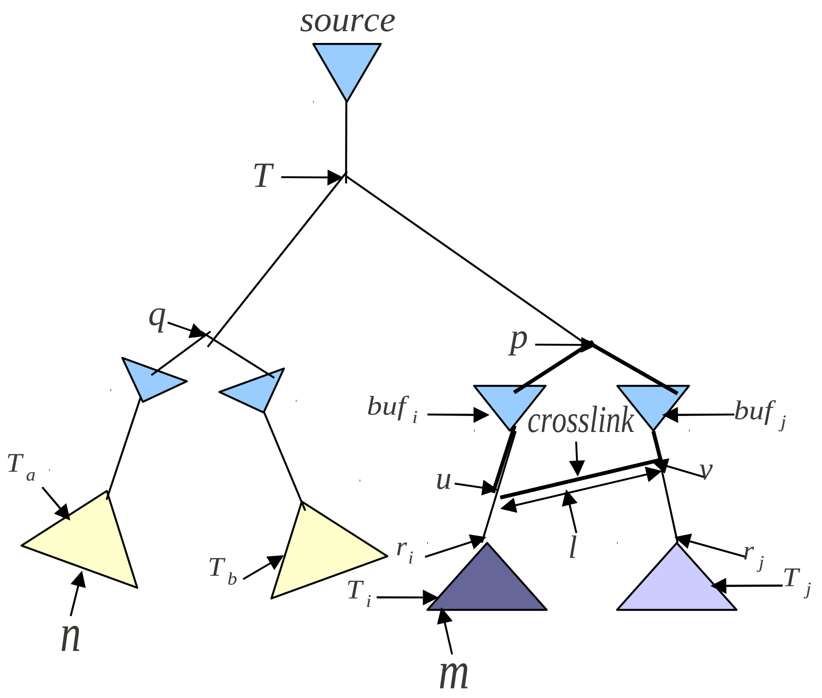


- Method 2:
  - Same delays for sinks within a sub-tree
  - Uniform correlation between all sink pairs m and n



Sinks are in two disjoint sub-trees

- No predictable correlation between delays of sinks m and n due to no overlap path
- Both Method 1 and Method 2 are equally ineffective in this situation.



Clock Network Synthesis

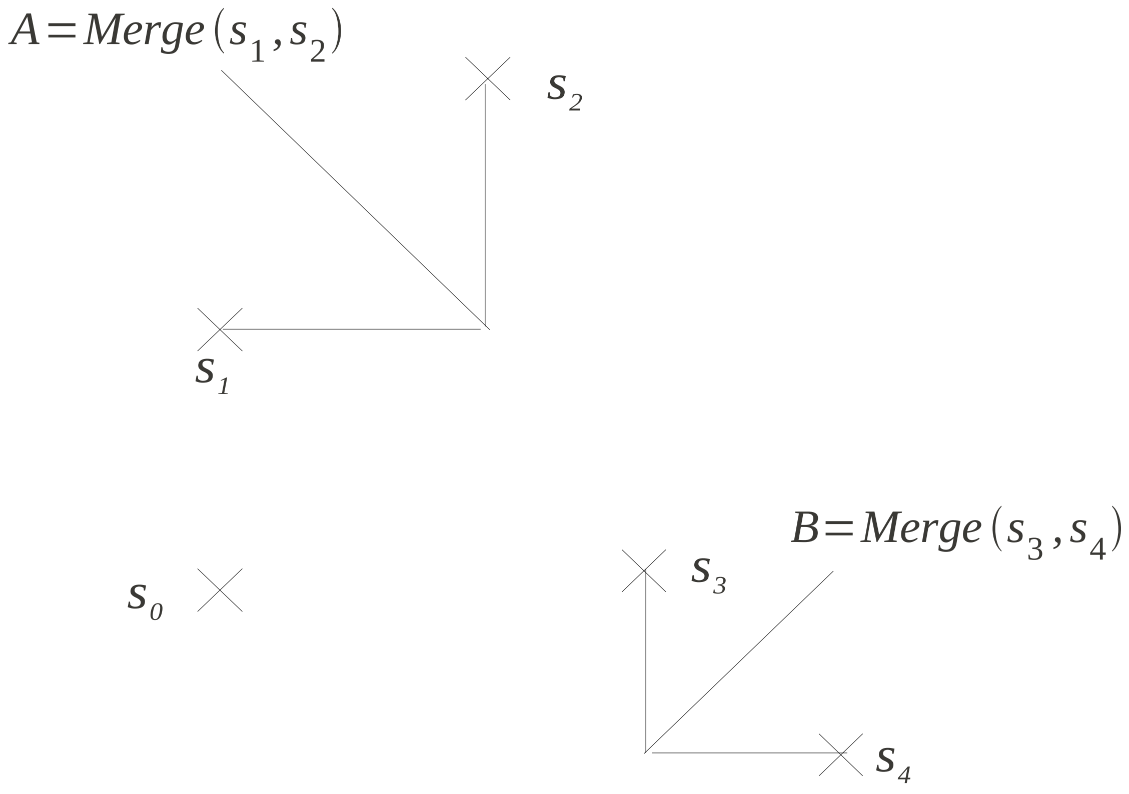
- Our clock network synthesis is based on the usage of Method 2 for cross link insertion.
- Problem formulation is based on ISPD'10 High performance Clock Network Synthesis contest.
- Our approach to clock network synthesis consists of 3 main steps
    - Merging
    - Buffer Insertion
    - Link Insertion

Problem Formulation

- Given: Sinks, Blockages and clock source location
- Objective: Generate a clock network T that connects clock source to the sinks.
- Constraints:
  - All sink pairs with distance between them less than user specified distance are called local sink pairs.
  - All local sink pairs should satisfy Local clock skew constraint (LCS).
  - Slew at any point should be less than predefined limit S.
  - Buffers should not be placed in the blockages

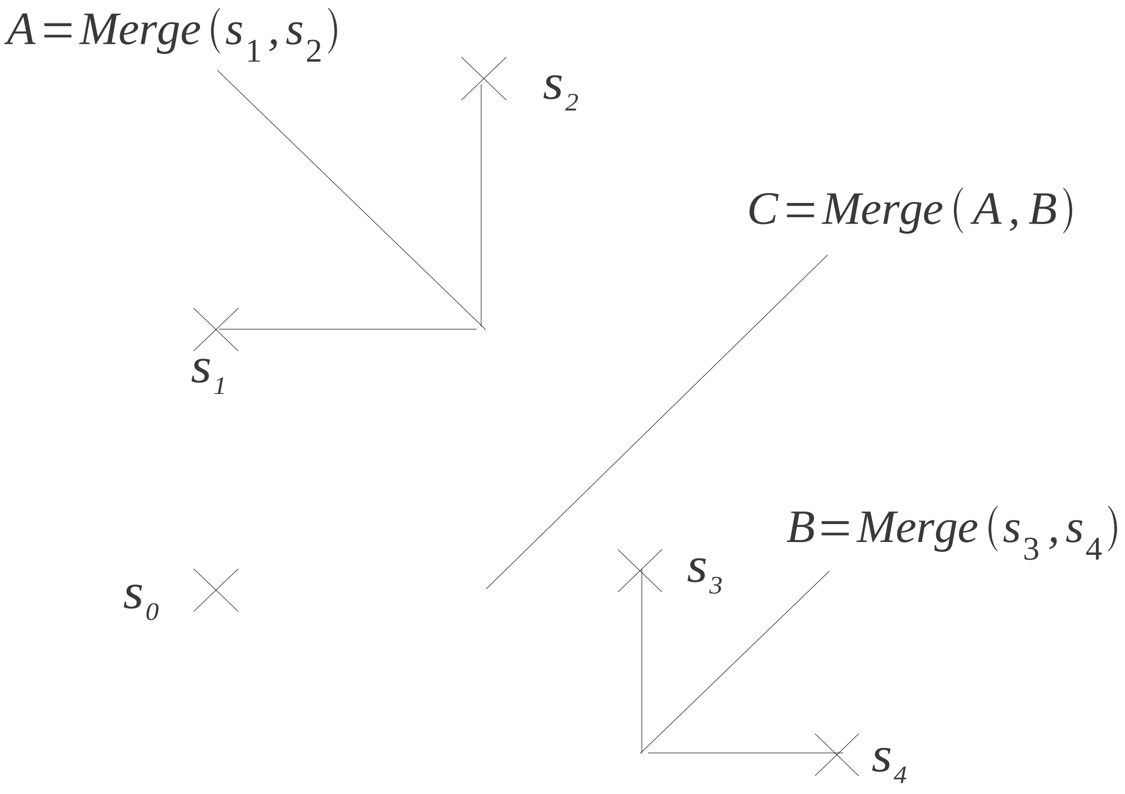
Merging

- General framework of Clock network synthesis is based on the Deferred-Merge embedding approach



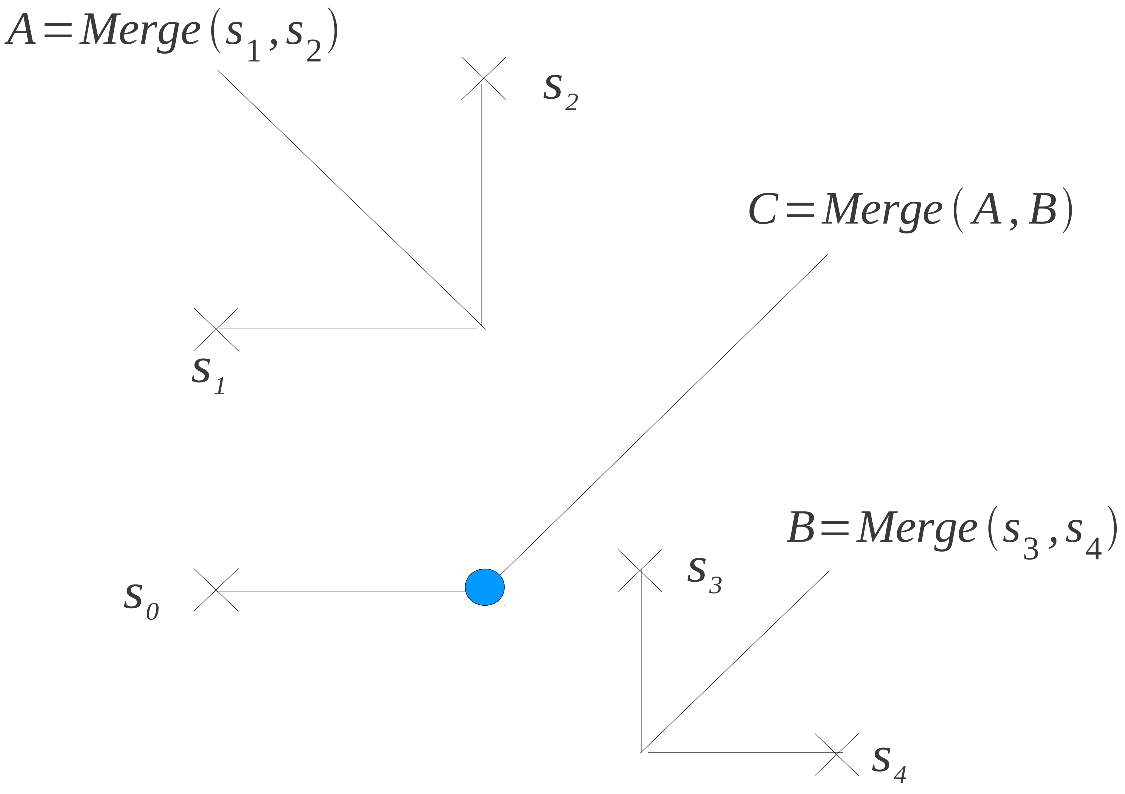
Merging

- General framework of Clock network synthesis is based on the Deferred-Merge embedding approach



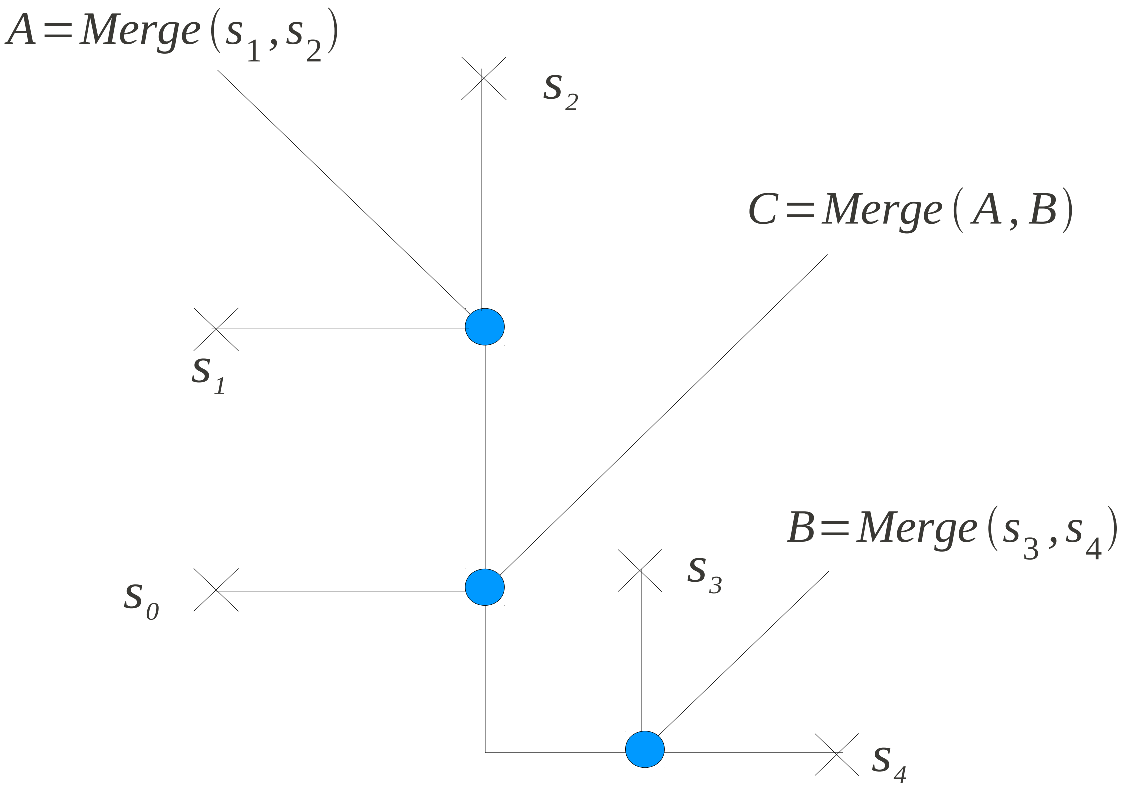
Merging

- General framework of Clock network synthesis is based on the Deferred-Merge embedding approach



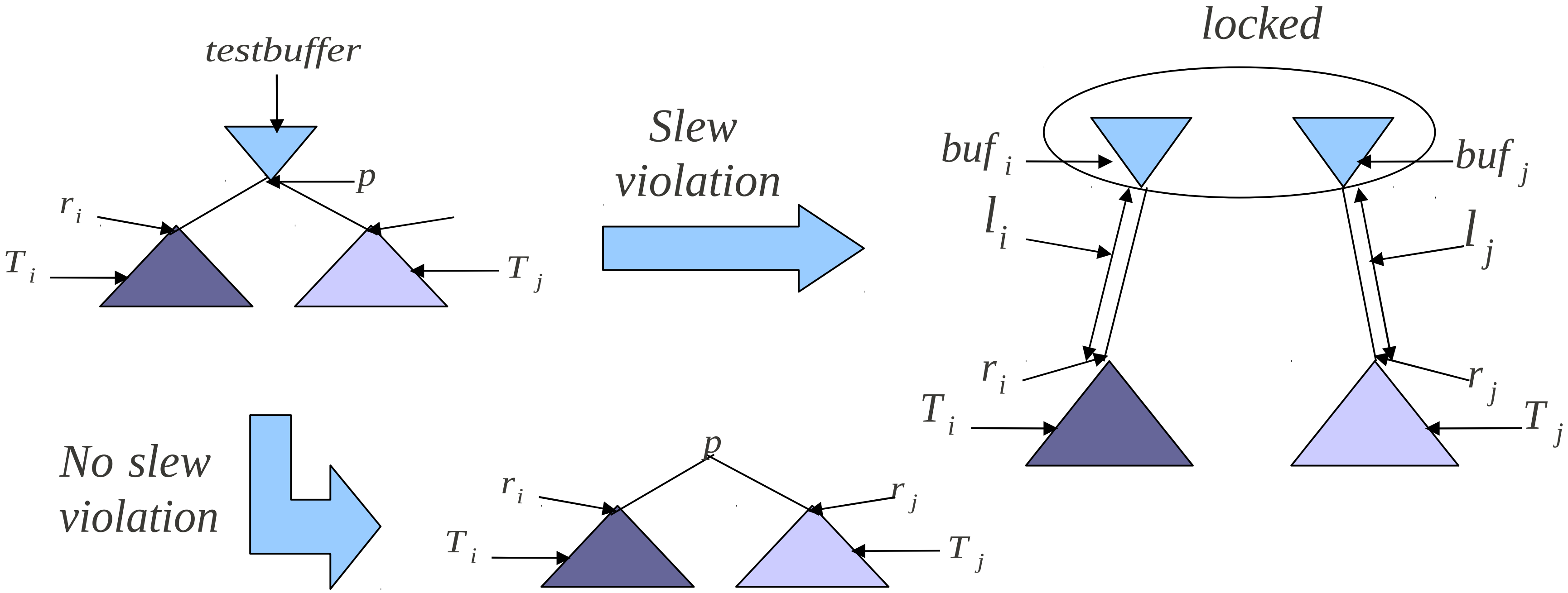
Merging

- General framework of Clock network synthesis is based on the Deferred-Merge embedding approach



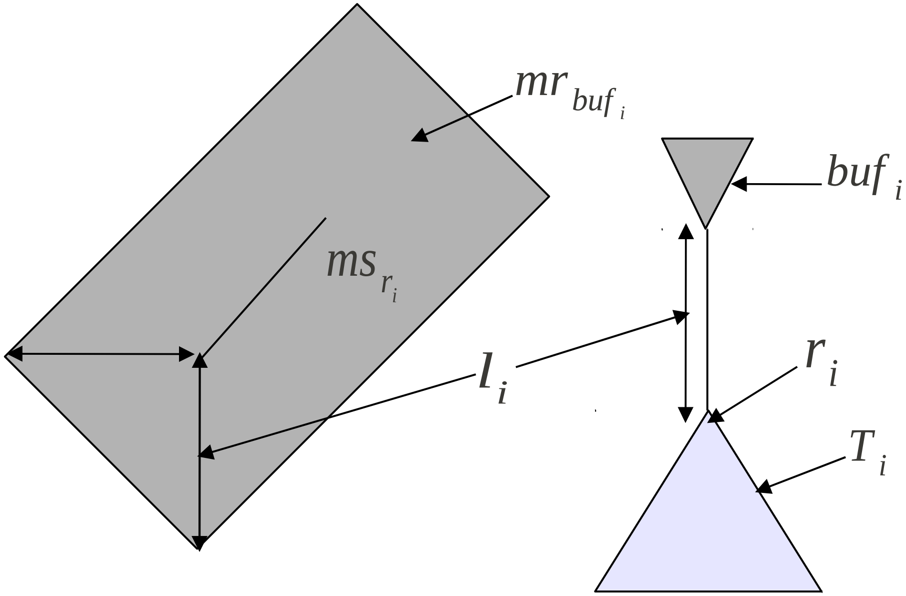
Merging

- In bottom-up phase clock tree is constructed iteratively.



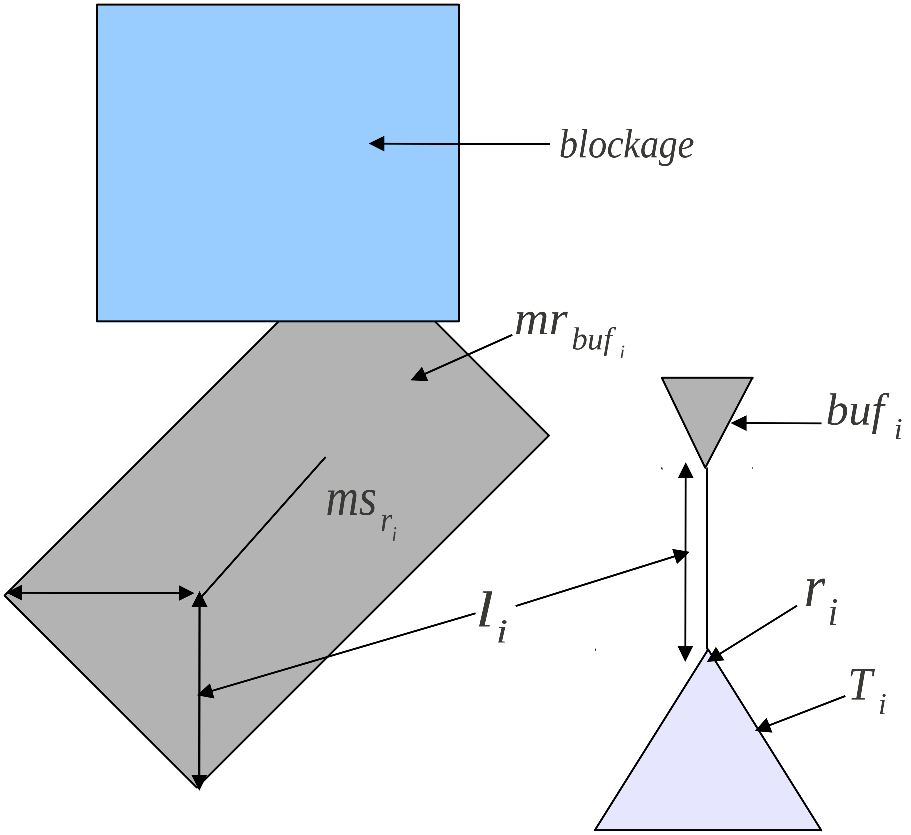
Buffer Insertion

- Slew constraints results in the buffer insertion in clock tree.
- Buffers are inserted on the stem wires.
- NGSPICE simulations are used to compute the length of stem wire.
- Each buffer buf_i has a merging region mr_{buf_i} associated with it.

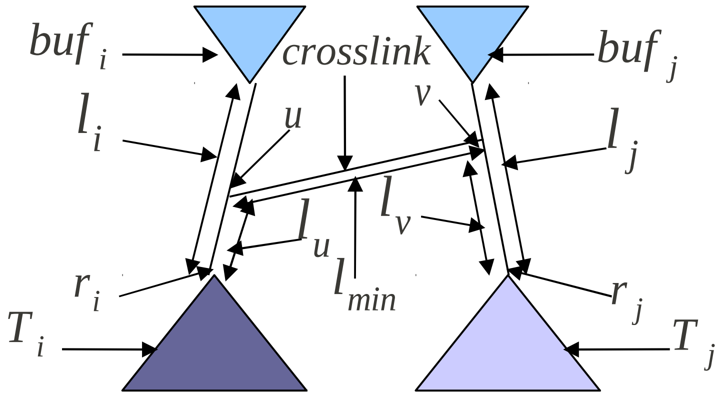


Buffer Insertion

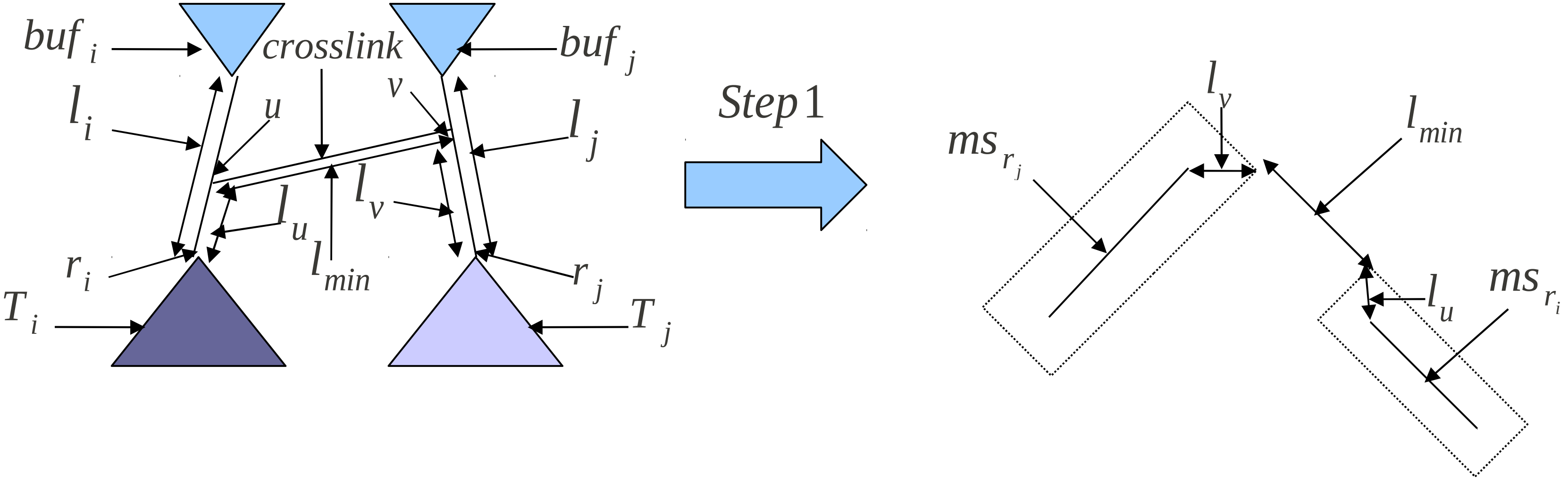
- Slew constraints results in the buffer insertion in clock tree.
- Buffers are inserted on the stem wires.
- NGSPICE simulations are used to compute the length of stem wire.
- Each buffer buf_i has a merging region mr_{buf_i} associated with it.
- Blockage avoidance is considered



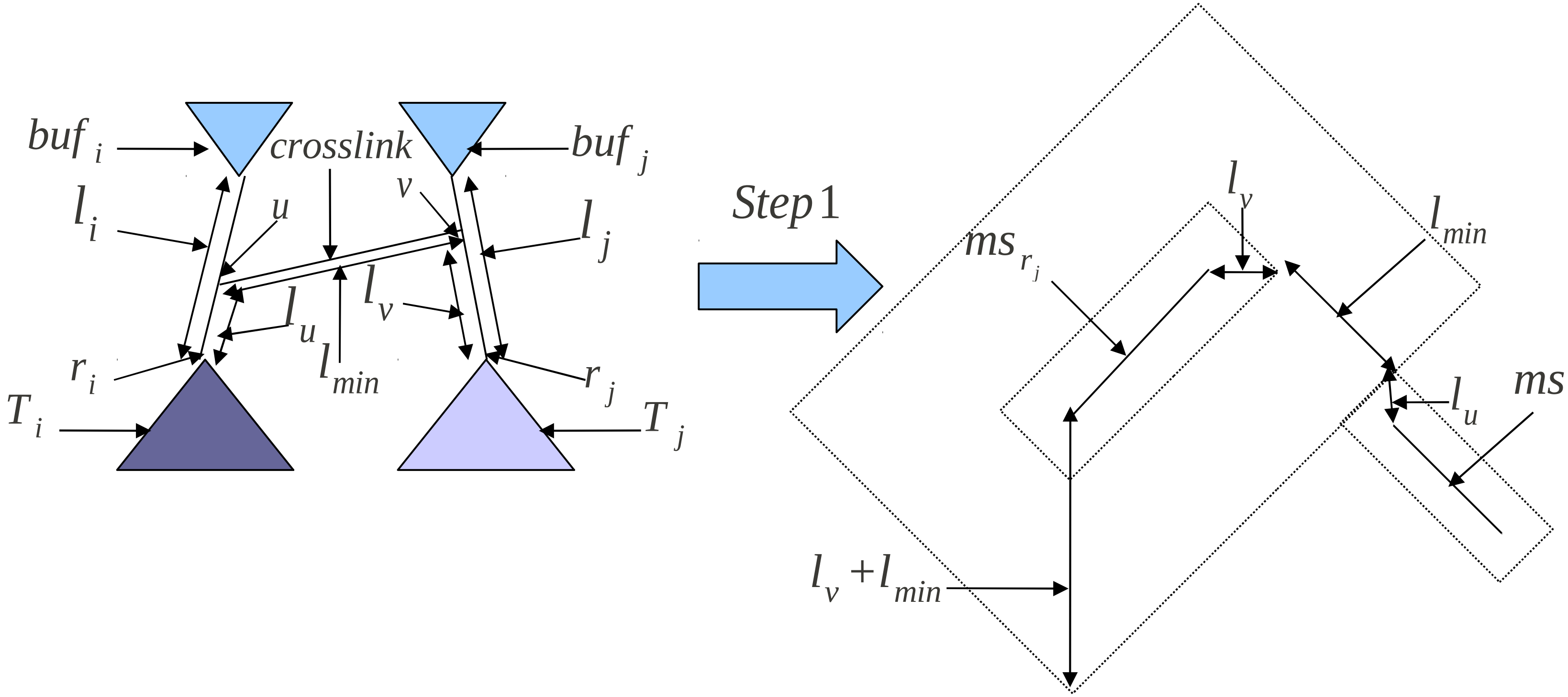
Link Insertion



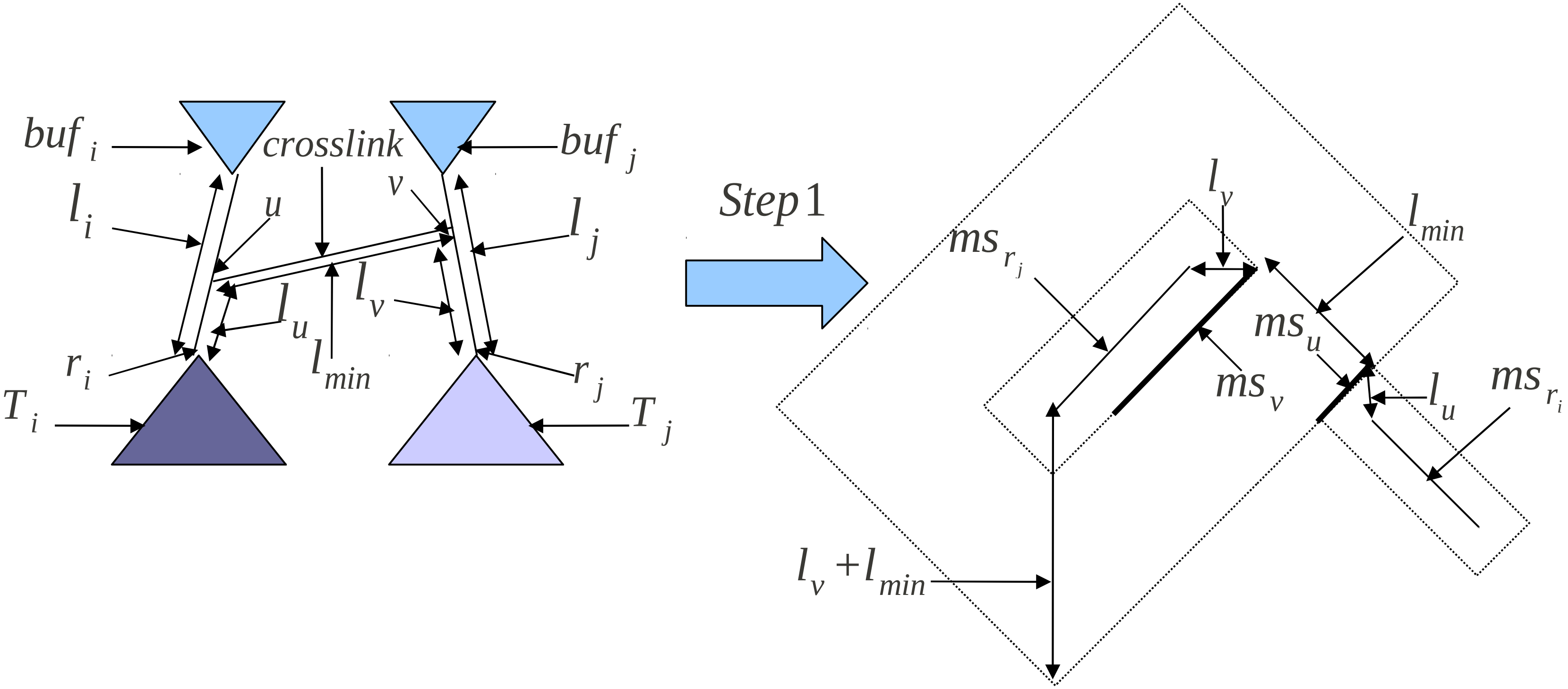
Link Insertion



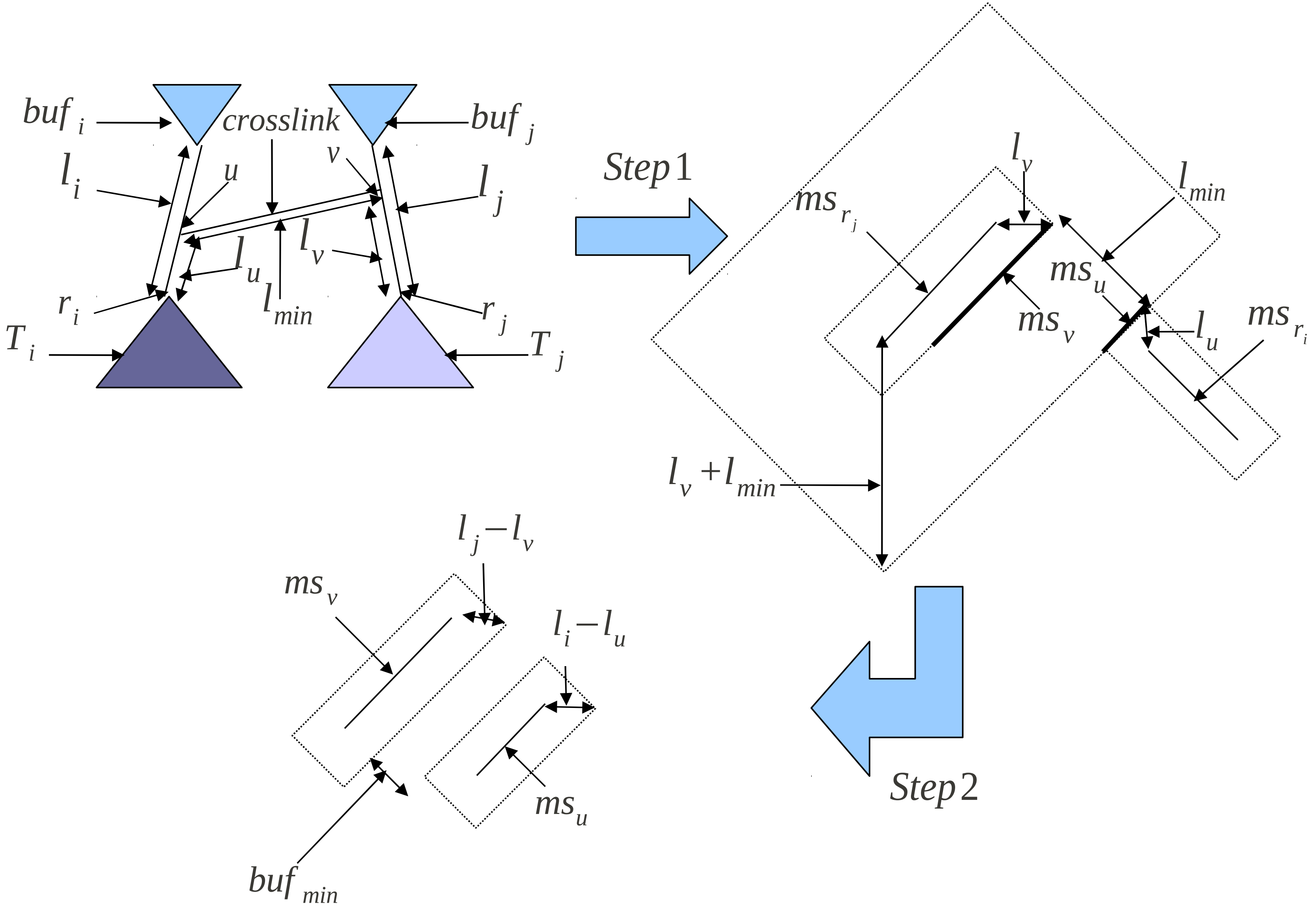
Link Insertion



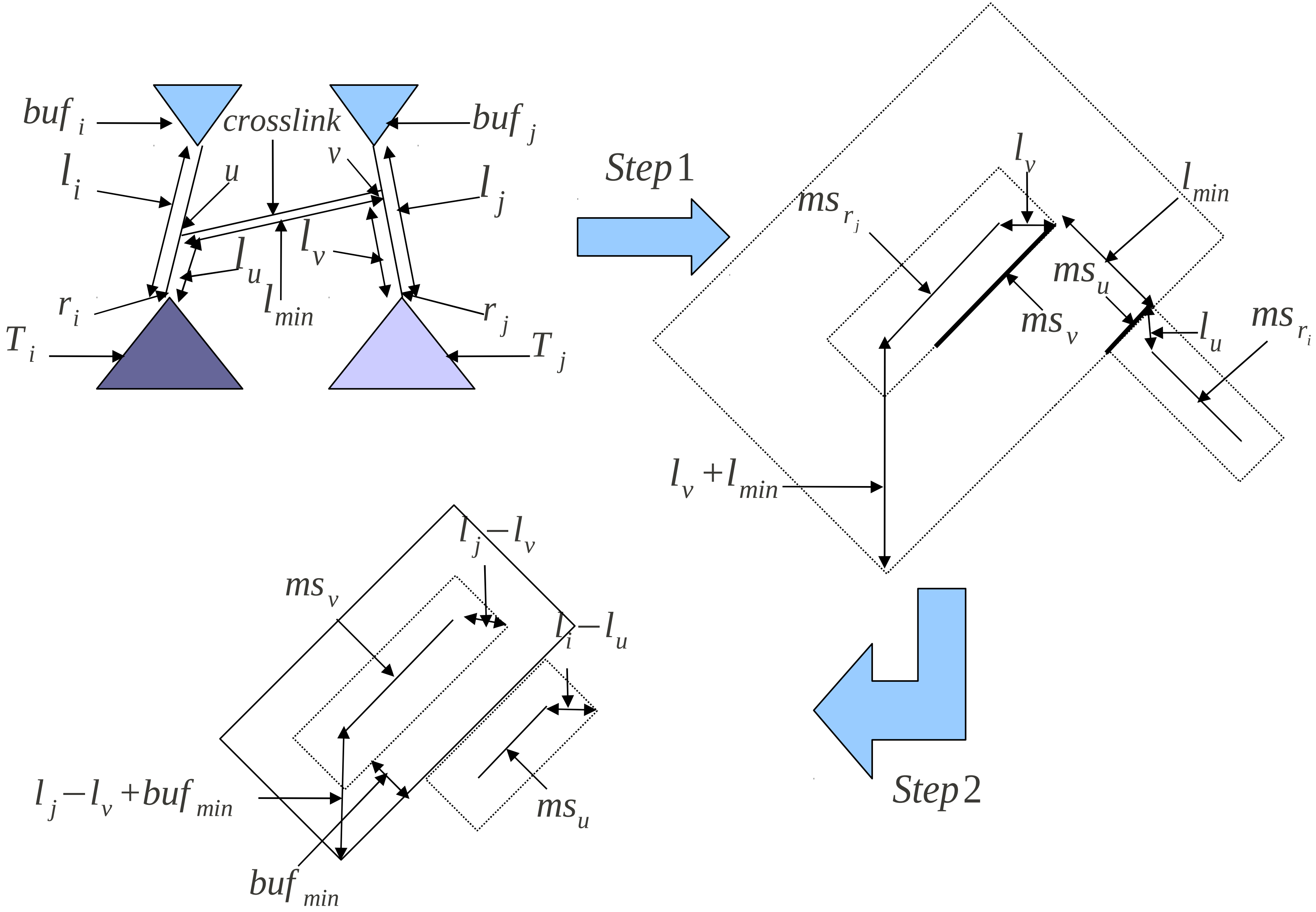
Link Insertion



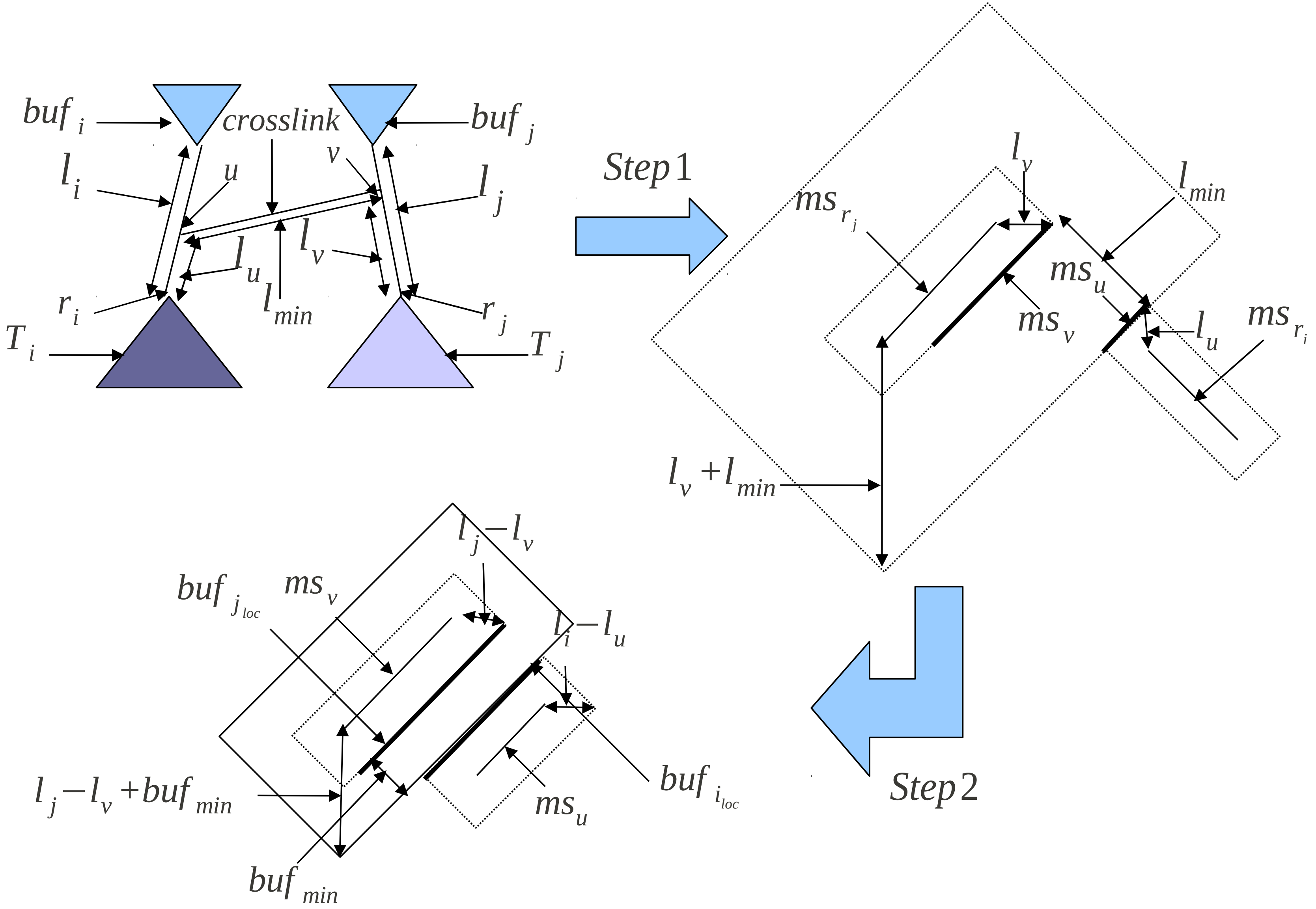
Link Insertion



Link Insertion



Link Insertion



Merits of our design flow

- Our link insertion flow allows us to control the link length.
- Inserting link below the buffer helps in reducing the variation effects of buffer as compared to inserting above it.
- Cross link maximizes the reduction of the skew variability for the sinks in the same sub-tree
- Cross link improves the correlation of the sink delays in the two sub-trees that are connected by the cross link.

Experimental Setup

- 45nm Predictive Technology Model
- Inverters types

  - Mid sized inverter (inv-1)

    - 10µm nmos, 14.6µm pmos (for similar R/F delay)
    - input cap=35fF, resistance=61.2Ω, output parasitic cap=80fF

  - Small inverter(inv-2)

    - 1.37µm nmos, 2µm pmos
    - input cap=4.2fF, resistance=440Ω, output parasitic cap=6.1fF

- Wire types

  - wire-1: 0.1(Ω/µm), 0.2(fF/µm)
  - wire-2: 0.3(Ω/µm), 0.16(fF/µm)

Experiment Setup

- Supply voltage variations=15%
- Wire width variations=10%
- Inverter size: 30 parallel inv-2
- Buffer size: 10 parallel inv-2 driving 40 parallel inv-2
- In ISPD Monte-Carlo simulations, each inverter gets supply voltage independent of other inverters in the circuit

Benchmark summary

Name	# sinks	LCS distance (nm)	LCS (ps)	Width (nm)	Height (nm)	# blockages
ispd10cns01	1107	600000	7.50	8000000	8000000	4
ispd10cns02	2249	600000	7.50	13000000	7000000	1
ispd10cns03	1200	370000	4.99	3071928	492989	2
ispd10cns04	1845	600000	7.50	2130492	2689554	2
ispd10cns05	1016	600000	7.50	2318787	2545448	1
ispd10cns06	981	600000	7.50	1949600	890880	0
ispd10cns06	1915	600000	7.50	2536640	1447680	0
ispd10cns08	1134	600000	7.50	1837440	1628160	0

ISPD Monte-Carlo Simulations

BM	# sinks	LCS (ps)	Method	95% LCS (ps)	Cap (fF)	Cap ratio	CPU (s)
01	1107	7.50	Contango[1,18]	7.01	198337	1.44	12015
			CNSrouter[1,19]	7.23	1168104	8.52	675
			NTUclock[1]	8.66	293887	2.14	15
			Work in [20]	7.16	445331	3.25	0.40
			Our work (buf)	7.32	142325	1.03	1092
			Our work (inv)	7.03	136961	1.00	3237
02	2249	7.50	Contango[1,18]	7.34	375863	1.48	25006
			CNSrouter[1,19]	7.35	2099811	8.27	2140
			NTUclock[1]	10.73	832483	3.28	176
			Work in [20]	7.33	933574	3.67	2.42
			Our work (buf)	7.42	263198	1.03	4314
			Our work (inv)	7.36	253760	1.00	10157
03	1200	4.99	Contango[1,18]	4.18	55861	1.51	3840
			CNSrouter[1,19]	3.95	93965	2.54	21
			NTUclock[1]	8.63	167062	4.53	6
			Work in [20]	4.88	183702	4.98	1.57
			Our work (buf)	4.49	36609	0.99	383
			Our work (inv)	4.82	36867	1.00	1761

ISPD Monte-Carlo Simulations contd...

BM	# sinks	LCS (ps)	Method	95% LCS (ps)	Cap (fF)	Cap ratio	CPU (s)
04	1845	7.50	Contango[1,18]	4.46	71843	1.51	6075
			CNSrouter[1,19]	7.25	125333	2.64	22
			NTUclock[1]	9.55	325206	6.86	58
			Work in [20]	4.09	196337	4.14	0.27
			Our work (buf)	6.70	51070	1.07	934
			Our work (inv)	6.79	47393	1.00	2543
05	1016	7.50	Contango[1,18]	4.41	37690	1.48	2406
			CNSrouter[1,19]	7.27	74084	8.27	10
			NTUclock[1]	6.98	130389	3.28	11
			Work in [20]	3.81	89094	3.67	0.40
			Our work (buf)	4.78	25129	1.03	278
			Our work (inv)	4.41	22589	1.00	778
06	981	7.50	Contango[1,18]	6.05	47810	1.63	2660
			CNSrouter[1,19]	6.79	87390	2.98	41
			NTUclock[1]	416.62	2E+06	68.31	1
			Work in [20]	7.49	160447	5.48	0.28
			Our work (buf)	6.41	32680	1.11	285
			Our work (inv)	5.81	29278	1.00	995

ISPD Monte-Carlo Simulations contd...

BM	# sinks	LCS (ps)	Method	95% LCS (ps)	Cap (fF)	Cap ratio	CPU (s)
07	1915	7.50	Contango[1,18]	4.58	72644	1.52	2351
			CNSrouter[1,19]	5.97	128351	2.69	27
			NTUclock[1]	8.12	275597	5.79	66
			Work in [20]	6.24	228243	4.79	0.30
			Our work (buf)	5.86	48316	1.01	818
			Our work (inv)	5.53	47555	1.00	2765
08	1134	7.50	Contango[1,18]	5.15	52490	1.68	1987
			CNSrouter[1,19]	5.37	97421	3.13	17
			NTUclock[1]	7.64	165883	5.33	7
			Work in [20]	5.47	228243	7.34	0.28
			Our work (buf)	5.07	33029	1.06	367
			Our work (inv)	5.72	31088	1.00	938

- We were able to meet the LCS constraint for all benchmarks with lower capacitance as compared to previous work.

Conclusions and Future Work

- Conclusions
  - New link insertion methodology of inserting links between higher level internal nodes in a clock tree is proposed
  - Proposed methodology improves the correlation of sink delays for the sinks that have similar path lengths to the inserted cross link
  - NGSPICE based Monte-Carlo simulations verifies the effectiveness of the approach
- Future work
  - Merging to minimize the local clock skew instead of global skew
  - Handling of longer cross links

Thank You